

ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОДЕЛЬНЫХ СИСТЕМ ДОМЕННОГО ЦЕХА

Современные реалии разработки программного обеспечения показывают, что лидирующей оказалась парадигма объектно-ориентированного программирования [1]. Для качественной разработки программного продукта наряду с физическим моделированием необходимо выполнить объектное моделирование, которое будет положено в основу программного продукта. В докладе показан пример объектного моделирования бизнес-процессов, реализованный в ходе проектирования структуры программного обеспечения оценки и прогнозирования шлакового режима доменной плавки в системе анализа и прогнозирования производственных ситуаций доменного цеха, внедренной на ОАО «Магнитогорский металлургический комбинат» [2].

Основные объекты можно разделить на четыре базовых направления:

- компоненты (Components);
- составные части (Ingredients);
- наложение (Mapping);
- параметры (Performance).

В соответствии этим направлениям в среде программирования Microsoft Visual Studio.NET (C#) были созданы соответствующие пространства имен (namespace).

1. *Пространство имен «Компоненты» (Components).* Класс Component моделирует химический состав элемента шихты. В основе класса Component лежит объект Dictionary<String, Double>, где содержится список параметров: название соединения или химического элемента (например, CaO, SiO₂, MgO и др.) и его величина. Все компоненты разделены на два подкласса:

- компоненты, загружаемые в печь (ComponentIn):
 - флюсы (ComponentFlux);
 - кокс (ComponentCoke);
 - ЖРМ (ComponentGRM);
- компоненты жидких продуктов плавки (ComponentOut):
 - чугун (ComponentIron);
 - шлак (ComponentSlag).

На рис. 1 показана диаграмма наследования типа **Component**.

Помимо химического состава каждый компонент содержит два обязательных поля:

- имя (**Name**) – содержит информацию об источнике, например, «Руда Михайловская»;
- расход (**Debit**) – для загружаемых компонентов;
- выход (**Credit**) – для получаемых компонентов.

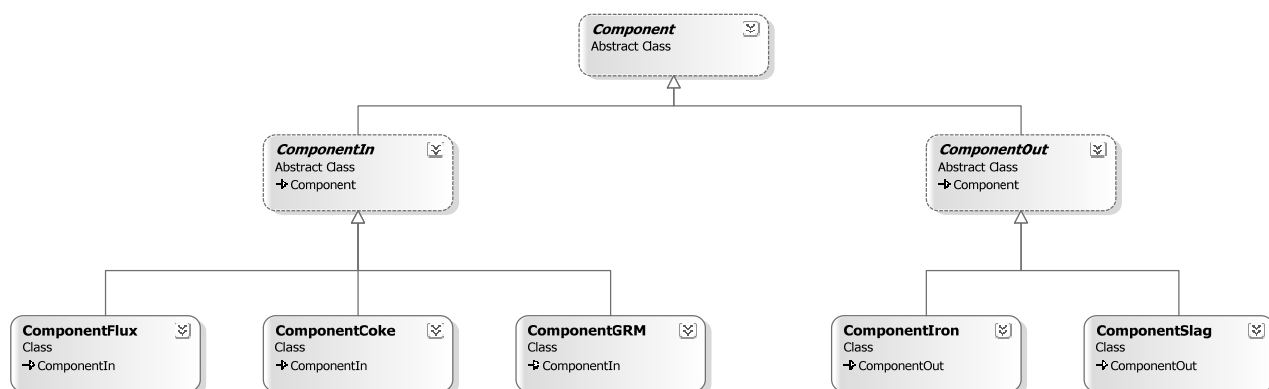


Рис. 1. Диаграмма наследования типа **Component**

Поля «Расход» и «Выход» необходимы для проведения математических расчетов.

2. *Пространство имен «Составные части» (Ingredients).* Предполагается, что составные части состоят из компонентов, поступающих в печь. Поэтому в основе класса лежит объект **List<ComponentIn>**. Здесь можно выделить три основных типа:

- ЖРМ (GRM);
- кокс (Coke);
- флюсы (Flux).

Каждая из этих сущностей содержит список соответствующих ей компонентов, например, GRM содержит список **ComponentGRM**. На рис. 2 продемонстрирована диаграмма наследования типа **Ingredient**.

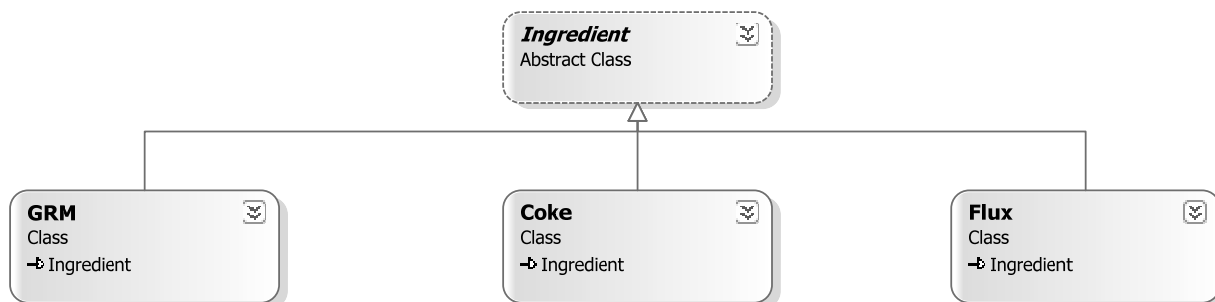


Рис. 2. Диаграмма наследования типа **Ingredient**

3. *Пространство имен «Наложение» (Mapping)*. Данное пространство имен содержит классы, главной задачей которых является взаимодействие с источником данных (базой данных), в результате которого должны быть заполнены экземпляры всех объектов: ЖРМ, кокса, флюсов, чугуна и шлака. В частности, должны быть получены фактические отчетные данные о химическом составе этих компонентов, а также ряд дополнительных параметров, необходимых для расчета (например, расходы компонентов, поступающих в печь).

В пространство имен «Наложение» входит два класса:

- Mapper;
- ComponentAttributes.

Mapper – это статический класс, который непосредственно взаимодействует с источником данных. Данный тип является универсальным способом доступа к источнику данных, который в ходе работы программы реализует один из базовых принципов объектно-ориентированного программирования – полиформизм, позволяя в зависимости от ситуации использовать наиболее удобный метод доступа к данным.

Второй метод ComponentAttributes является вспомогательным классом, который является своего рода проекцией на таблицу из базы данных. Он описывает поля, характеризующие компоненты шихты: код материала; имя материала; тип материала. В базе данных доменного цеха использованы следующие номенклатурные типы материалов: агломерат, окатыши, флюсы, металлдобавки, ЖРМ, топливо, зола, отходы.

4. *Пространство имен «Параметры» (Performance)*. Данное пространство имен содержит классы, которые моделируют параметры компонентов. Все параметры разделены на два подкласса:

- технические параметры (TechPerformance) – отчетные (ReportTechPerformance) и сравниваемые (CompareTechPerformance);
- параметры расхода (DebitPerformance) – отчетные (ReportDebitPerformance) и сравниваемые (CompareDebitPerformance).

На рис. 3 проиллюстрирована диаграмма наследования типа Performance.

Необходимо также использовать тип DebitValue, который содержит три поля:

- наименование;
- дата;
- величина.

Типы классов с префиксом Report содержат по одному экземпляру DebitValue, в которых хранится отчетная величина параметра. В типах с префиксом Compare к существующему параметру Report добавляется еще один параметр – Compare, в котором содержатся данные в сравнительном

периоде. Кроме указанных элементов в объектах наследниках от CommonPerformance содержится еще ряд дополнительных параметров:

- имя Name – имя параметра;
- годовой IsYear – параметр за год;
- отладочный IsDebug – отладочный вызов;
- номер печи PechNum.

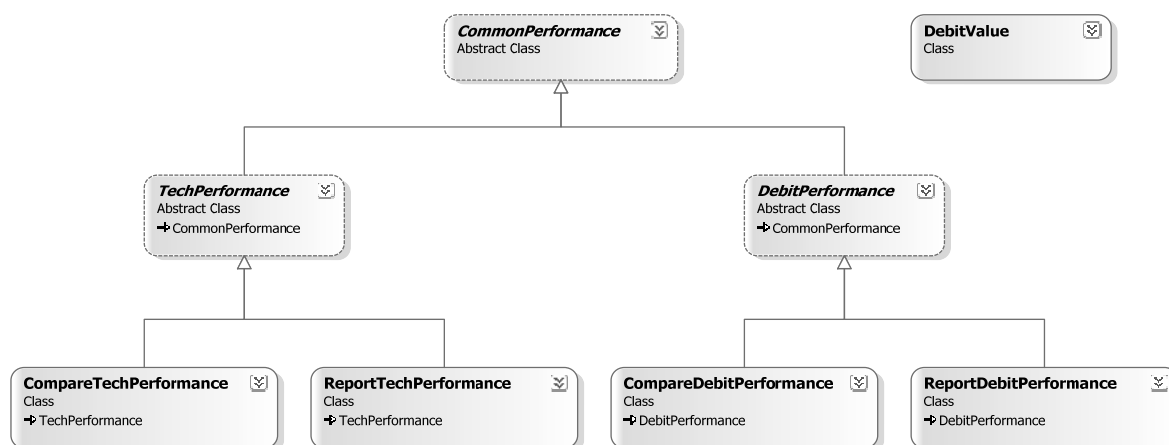


Рис. 3. Диаграмма наследования типа Performance

Эффективность разработки программного обеспечения во многом можно оценить по тому, насколько верно выбраны объекты, которые подлежат объектному моделированию. Часто именно эти же сущности ложатся в основу базы данных. По этой причине существует возможность автоматизировать процесс заполнения (создания) экземпляров типов в программе из базы данных. Для этих целей используются ORM (Object Relation Mapping) системы.

Таким образом, показанные направления Components, Ingridients, Mapping и Performance в виде классов реализуют всю необходимую функциональность программного модуля оценки и прогнозирования шлакового режима доменной плавки. Итоговый модуль реализован в виде .Net сборки, скомпилированной для операционной платформы Windows.

Список использованных источников

1. Макконнелл С. Профессиональная разработка программного обеспечения. СПб.: Символ плюс, 2007. 240 с.
2. Модельные системы поддержки принятия решений в АСУ ТП доменной плавки / Спирин Н.А., Лавров В.В., Рыболовлев В.Ю. [и др.]. Екатеринбург: УрФУ, 2011. 456 с.